Security with NaturalX

This section covers the following topics:

- Overview
- Activation Security
- Call Security

Information on how to configure NaturalX is given in the section Configuration Examples.

Overview

In a distributed environment, security is an especially important topic. A server must be sure that no unauthorized clients use the services it provides. A client must be sure that it is connected to the server it expects, and that the server does not misuse its (the client's) authorizations.

In the context of DCOM, two levels of security can be distinguished:

- Activation security controls who is allowed to launch and access the server process that provides the class.
- Call security controls who is allowed to use the individual methods a class provides.

In many cases, activation security may be sufficient to define authorizations. This security level is supported by DCOM itself on the basis of Windows Security. The necessary authorizations are maintained in the system registry. This is described in the section Activation Security.

In other cases it may be necessary to control authorizations in more detail at the level of individual methods. This security level cannot be maintained with registry definitions. It is, therefore, provided by NaturalX with the help of Natural Security. This is described in the section Call Security.

Activation Security

This section covers the following topics:

- Applications
- Authorizations using the Registry

Applications

Activation security controls who is allowed to launch and access a server process. In principle, this could be done by defining authorizations for each individual class. For practical reasons, however, and to reduce administration efforts, authorizations are normally maintained at the application level. In the system registry, each application is defined by an AppID. The AppID is the key under which the authorizations for an application are maintained. To maintain these authorizations, each DCOM enabled platform provides the tool DCOMCNFG. This tool can be used for NaturalX applications as well as for other DCOM applications.

In order to understand the meaning of AppIDs in NaturalX, recall for a moment how NaturalX organizes classes to applications (see the section Organizing Server IDs). With the Natural parameter COMSERVERID=*server-ID*, a name can be given to a certain NaturalX server. When Natural is started with a given value of COMSERVERID=*server-ID*, all Natural classes that are registered during this Natural session are registered under this server ID. At the same time, they are all registered under the same AppID key in the system registry. This means that each different value of *server-ID* corresponds to a different AppID key in the system registry.

As an example, assume Natural is running with the server ID "Employees". All classes registered during this Natural session will then form the "Employees" application. The REGISTER command registers them all under one AppID key - the one that corresponds to the "Employees" application.

Authorizations using the Registry

When configuring Activation Security, the following registry keys are of interest: *LaunchPermissions, AccessPermissions, DefaultLaunchPermissions* and *DefaultAccessPermissions*. The keys *DefaultLaunchPermissions* and *DefaultAccessPermissions* exist only once in the registry and define authorizations for all applications for which no individual authorizations have been defined. The keys *LaunchPermissions* and *AccessPermissions* exist for each application (i.e. for each AppID) and define the authorizations for an individual application.

Call Security

This section covers the following topics:

- Authorizations using Natural Security
- Security Hints and Suggestions

Authorizations using Natural Security

Call security is used to control who is allowed to use the individual methods that a class provides. Authorizations on this level cannot be maintained by registry definitions. Call security is therefore provided by NaturalX with the help of Natural Security.

In order to understand how call security is achieved with Natural Security, consider how a class in NaturalX is implemented: each class is a Natural module of type class, each method is a Natural module of type subprogram. For all Natural modules, the execution can be controlled by authorizations defined in Natural Security. Please refer to the *Natural Security Manual* for further information about how to do this.

The authorizations defined for class modules and method subprograms are evaluated whenever a class module is used to create objects and whenever a method subprogram is executed in response to a method call. The following rule applies: a user who is allowed to execute the class module is allowed to create objects of that class, and a user who is allowed to execute a method subprogram is allowed to use the corresponding method.

In order to perform the necessary authorization checks, a NaturalX server must know the client’s user ID. It must also be sure that the user ID is authentic. Therefore the following requirements must be met to use call security:

- The client must have identified itself with a logon on its local machine or on a Windows domain server.
- *Authentication level* must be set to at least *Connect* (either on the client or on the server machine).
- *Impersonation level* must be set to at least *Identify* (either on the client or on the server machine).

If the above requirements are met, a NaturalX server that is going to process a request takes the client’s user ID and places it into the Natural system variable *USER. The request is then performed under this user ID, including all necessary Natural Security authorization checks. After having processed the request, the Natural system variable *USER reverts to the value that it had at the startup of the NaturalX server.

If one of the requirements is not met, *USER remains unchanged during execution of the request. The request is then executed under the user ID under which the NaturalX server was started.

In addition to *USER, also the system variable *NET-USER is filled during execution of a request. It contains the user ID qualified with the domain name for clients belonging to a Windows domain and can be used for additional application-specific security checks.

Security Hints and Suggestions

The following points should be taken into consideration when using NaturalX with Natural Security:

- In a Natural Security environment, a NaturalX server must be started with the Natural parameter AUTO=ON. This is because the authentication already takes place on the client side. The setting should be entered in the Natural parameter module.
- In a Natural Security environment, it is a good idea to let a NaturalX server always start under a specific user ID. This user ID is then automatically used for all requests of unauthenticated users, and it is up to the Natural Security administrator to define minimal authorizations for this user ID.
- Remember that Natural and Natural Security cannot handle user IDs which are longer than 8 characters or which contain blanks.